实验项目一

一、程序代码

#include<iostream>

using namespace std;

class Coordinate {

private:

float Coord[100][100];

int times;

public:

Coordinate()

{

times = 2;

cout << "Coordinate construction1 called!" << endl;

}

Coordinate(int times1)

{

times = times1;

cout << "Coordinate construction2 called!" << endl;

}

~Coordinate()

{

cout << "Coordinate destruction called!" << endl;

}

void InputCoord()

{

for (int i = 0; i < times; i++)

{

cout << "Please Input x:" << endl;

cin >> Coord[i][1];

cout << "Please Input y:" << endl;

cin >> Coord[i][2];

}

}

void ShowCoord()

{

cout << "The coord is:" << endl;

for (int i = 0; i < times; i++)

{

cout << "(" << Coord[i][1] << "," << Coord[i][2] << ")" << endl;

}

}

void ShowAvgCoord()

{

float avgx = 0;

float avgy = 0;

for (int i = 0; i < times; i++)

{

avgx = avgx + Coord[i][1];

avgy = avgy + Coord[i][2];

}

avgx = avgx / times;

avgy = avgy / times;

cout << "The AVG coord is:" << endl;

cout << "(" << avgx << "," << avgy << ")" << endl;

}

};

int main()

{

Coordinate x;

x.InputCoord();

x.ShowCoord();

x.ShowAvgCoord();

Coordinate y(5);

y.InputCoord();

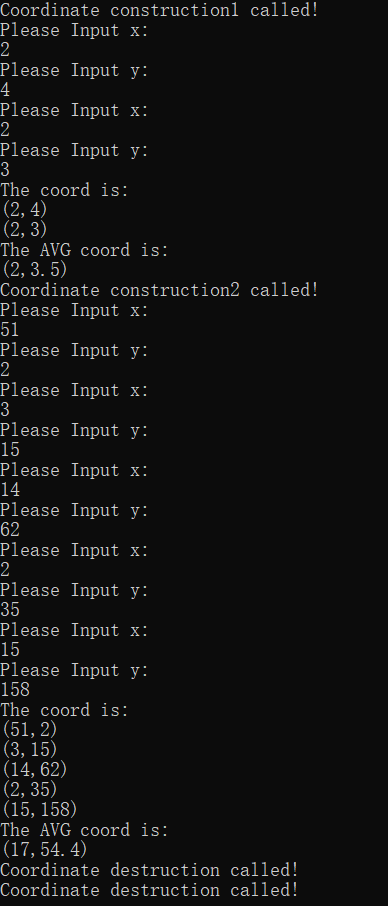
y.ShowCoord();

y.ShowAvgCoord();

return 0;

}

二、程序结果



三、感想心得

可以通过重载构造函数的方式实现构造函数接收数据的多样性

实验项目二

一、程序代码

#include <iostream>

#include<cstring>

using namespace std;

class Transcript {

private:

float grade[100][300];

int times;

char name[100][20];

float avg[100];

public:

Transcript(int times1);

void InputGrade();

void ShowGrade();

void ShowAvgGrade();

void sort();

};

Transcript::Transcript(int times1 = 2)

{

times = times1;

}

void Transcript::InputGrade()

{

for (int i = 0; i < times; i++)

{

cout << "请输入学生姓名：" << endl;

cin >> name[i];

cout << "请输入科目A成绩：" << endl;

cin >> grade[i][0];

cout << "请输入科目B成绩：" << endl;

cin >> grade[i][1];

cout << "请输入科目C成绩：" << endl;

cin >> grade[i][2];

}

}

void Transcript::ShowGrade()

{

for (int i = 0; i < times; i++)

{

cout << "姓名：" << name[i] << " 科目A成绩：" << grade[i][0] << " 科目B成绩:" << grade[i][1] << " 科目C成绩:" << grade[i][2] << endl;

}

}

void Transcript::ShowAvgGrade()

{

float avgA = 0;

float avgB = 0;

float avgC = 0;

for (int i = 0; i < times; i++)

{

avgA = avgA + grade[i][0];

avgB = avgB + grade[i][1];

avgC = avgC + grade[i][2];

avg[i] = (grade[i][0] + grade[i][1] + grade[i][2]) / 3;

}

avgA = avgA / times;

avgB = avgB / times;

avgC = avgC / times;

for (int i = 0; i < times; i++)

{

cout << "姓名：" << name[i] << " 平均成绩：" << avg[i] << endl;

}

cout << "课程名称：A 平均成绩：" << avgA << endl;

cout << "课程名称：B 平均成绩：" << avgB << endl;

cout << "课程名称：C 平均成绩：" << avgC << endl;

}

void Transcript::sort()

{

for (int k = 0; k < 3; k++)

{

for (int i = 0; i < times - 1; i++)

for (int j = 0; j < times - 1 - i; j++)

if (grade[j][k] < grade[j + 1][k])

{

float temp = grade[j + 1][k];

grade[j + 1][k] = grade[j][k];

grade[j][k] = temp;

char temp1[100][20];

strcpy\_s(temp1[0], name[j + 1]);

strcpy\_s(name[j + 1], name[j]);

strcpy\_s(name[j], temp1[0]);

}

}

cout << "课程名称：A" << endl;

for (int i = 0; i < times; i++)

{

cout << "姓名：" << name[i] << " 成绩：" << grade[i][0] << endl;

}

cout << "课程名称：B" << endl;

for (int i = 0; i < times; i++)

{

cout << "姓名：" << name[i] << " 成绩：" << grade[i][1] << endl;

}

cout << "课程名称：C" << endl;

for (int i = 0; i < times; i++)

{

cout << "姓名：" << name[i] << " 成绩：" << grade[i][2] << endl;

}

}

int main()

{

Transcript x(2);

x.InputGrade();

x.ShowGrade();

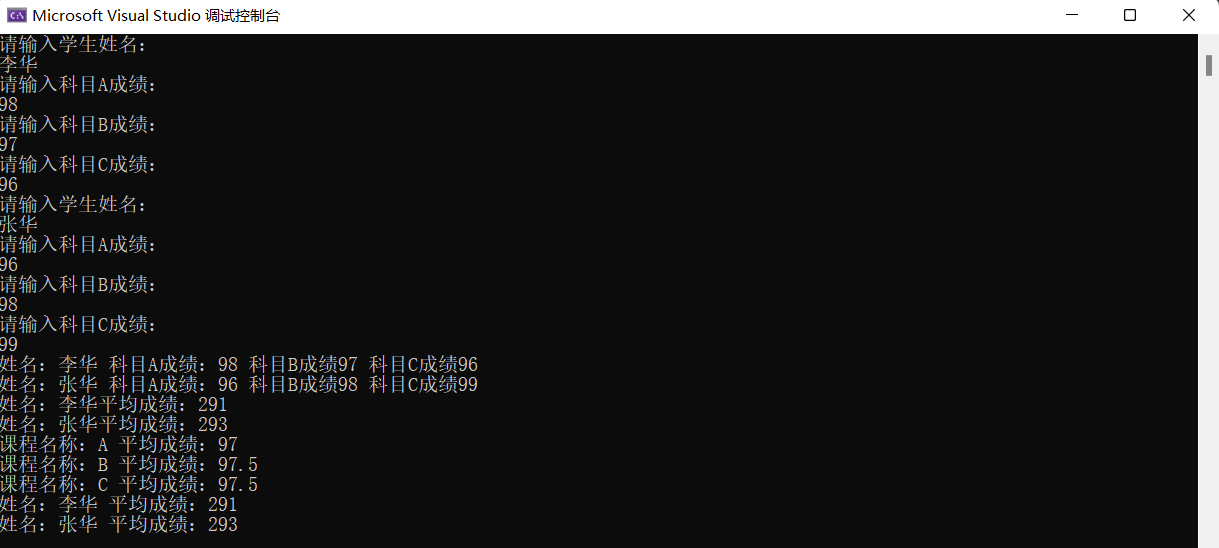
x.ShowAvgGrade();

x.sort();

return 0;

}

二、程序结果



三、感想心得

使用循环可以减少工作量。注意中文英文符号区别。

Copyright ©2021-2099 ShuoWang. All rights reserved